

# Pozor na výběr správného datového typu sloupce

TABLE\_A varchar(30) → varchar(255) → varchar(1000) → text

id	name	country
		"SK"
		"CZ"

char(2)

↖ X

# Primární a cizí klíč

Jméno Příjmení NČ Country ISO(2) Country Name

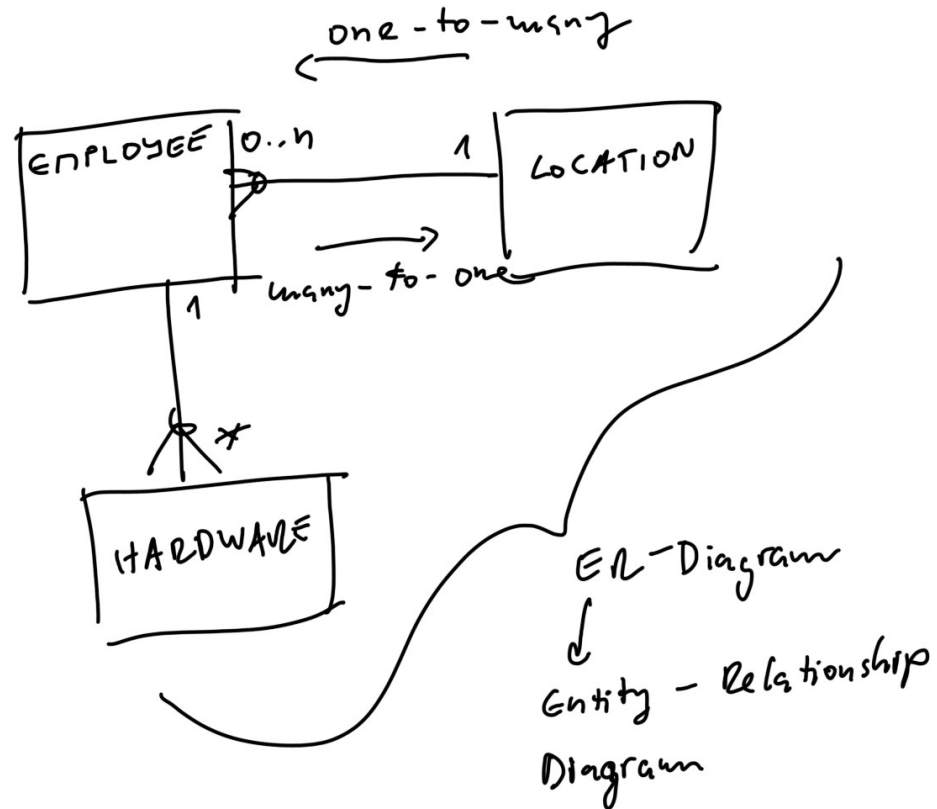
FK

User				
ID	F_NAME	L_NAME	NUM	COUNTRY_ID
bigserial PK				int
1	Jirka	P	xxx	1 - - -

Country		
ISO	NAME	ID serial PK
"CZ"	"Czech Rep"	1

PK = unique  
= NOT NULL

# ER diagram



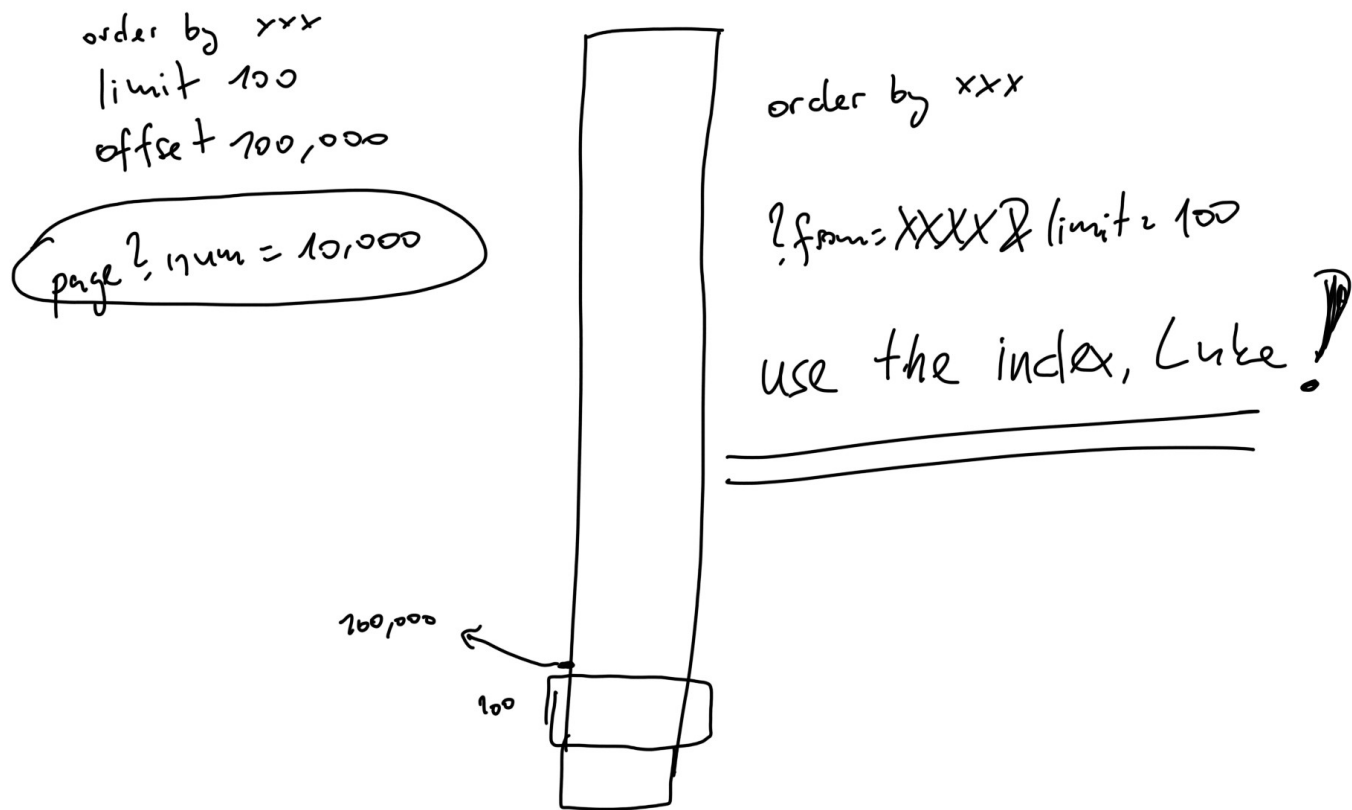
# Primární klíč

## PRIMÁRNÍ KLÍČ

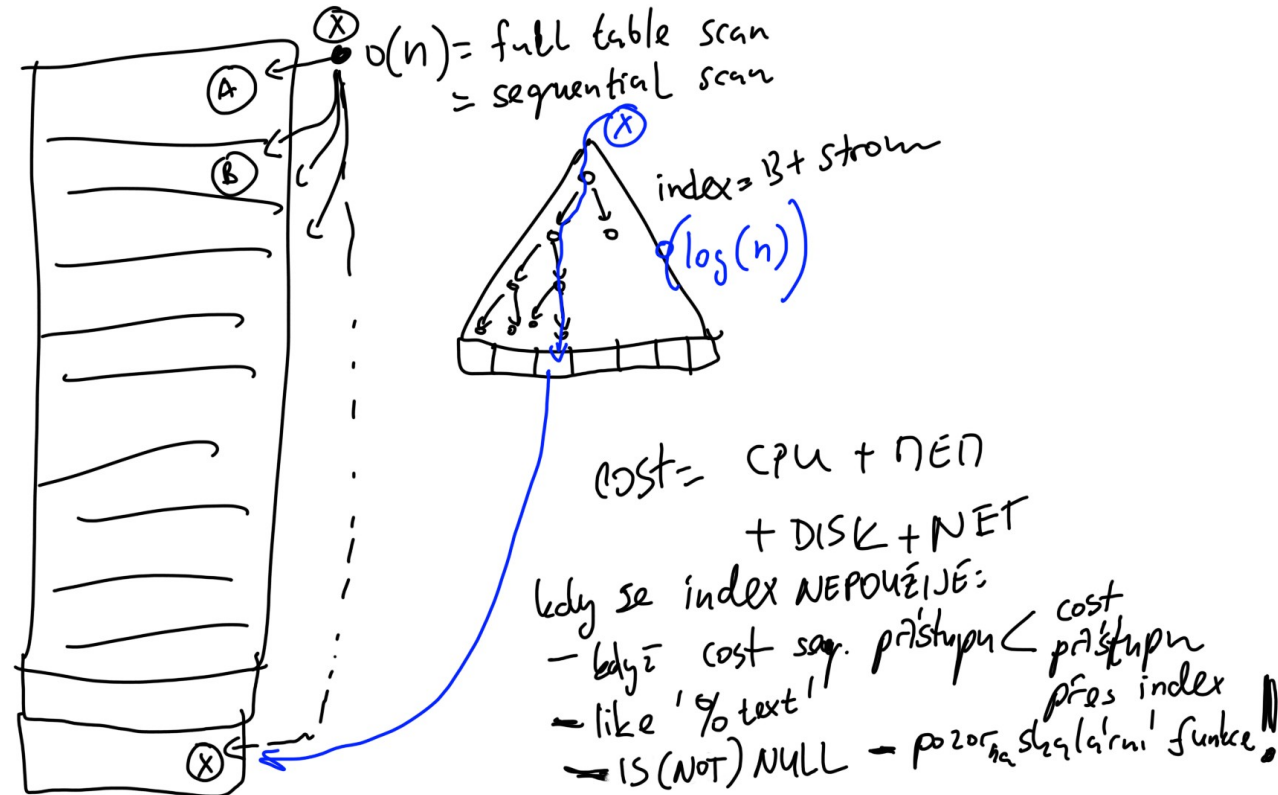
- každá tabulka musí mít PK
- PK = syntetický PK
  - id ze sequence ✓
  - uuid ✓
  - "cz" X
  - nc X

PK : FK by měl být stejnými hodnotami NIKDLI složený klíč

Pozor na OFFSET!!! Je to velice neefektivní způsob stránkování!!!



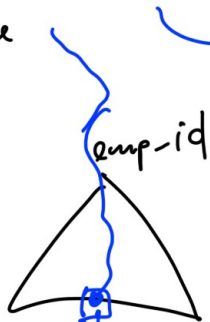
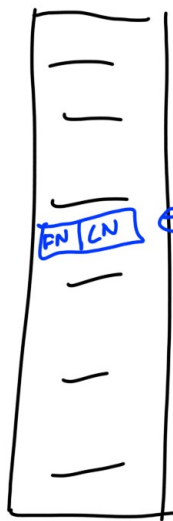
# Sekvenční průchod tabulkou vs. použití indexu



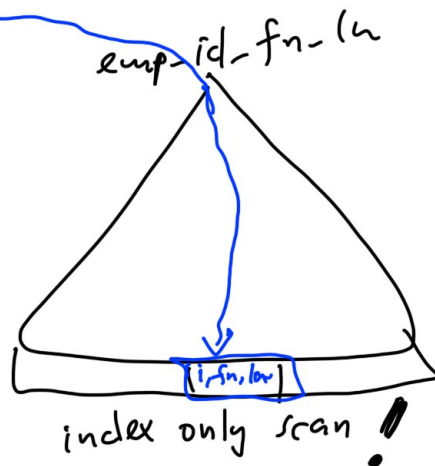
Kompozitní index má výhodu, že se dá vyhledávat pouze v něm a nemusí se ještě jít do tabulky.

select first\_name, last\_name  
from employee  
where id = ?

employee

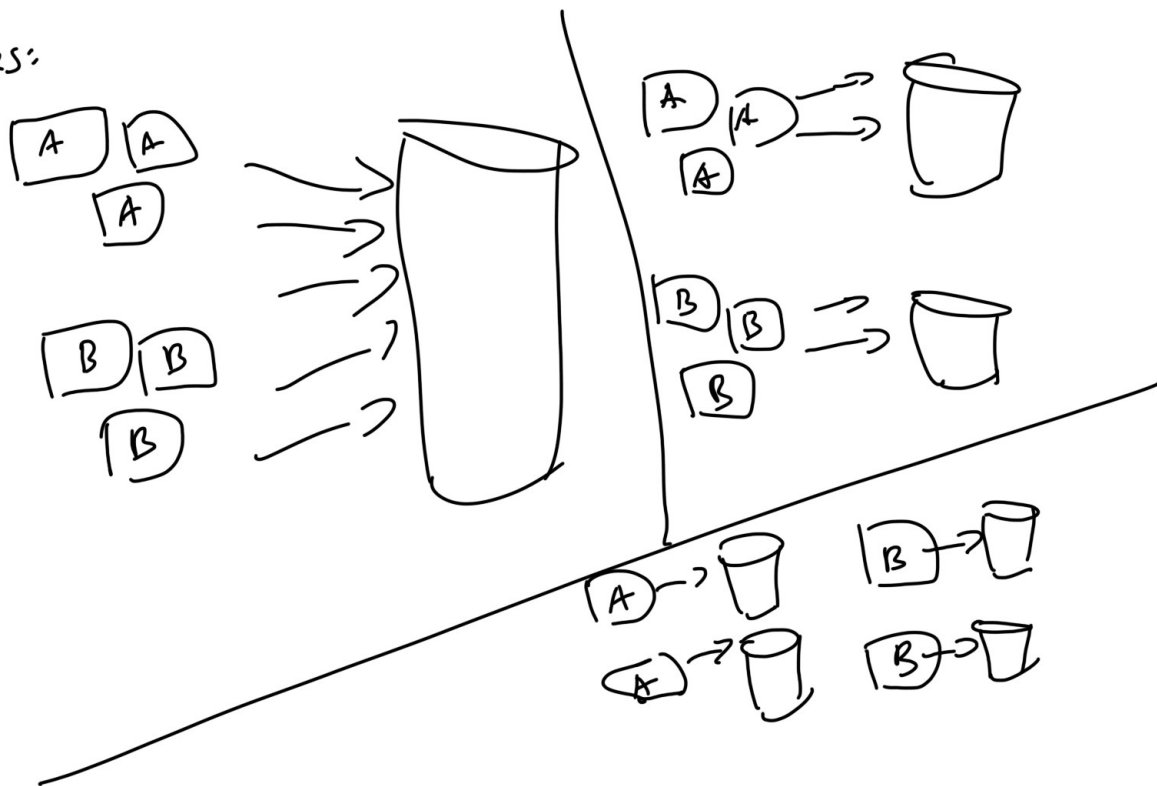


vs,



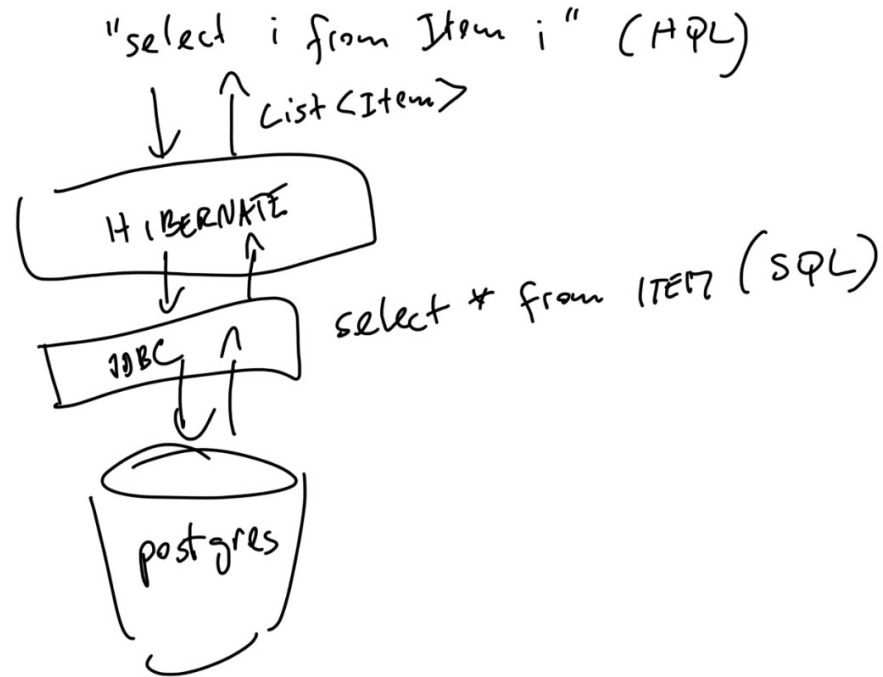
# Microservices a databáze

microservices:

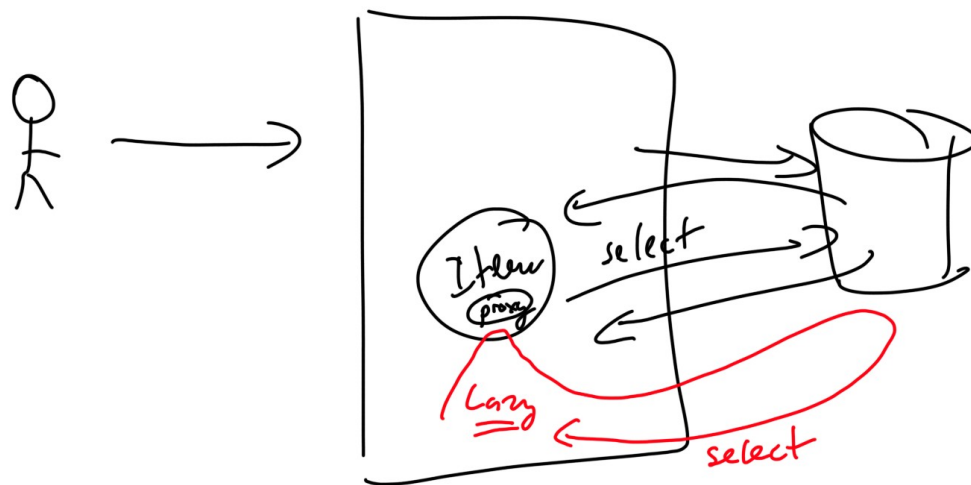




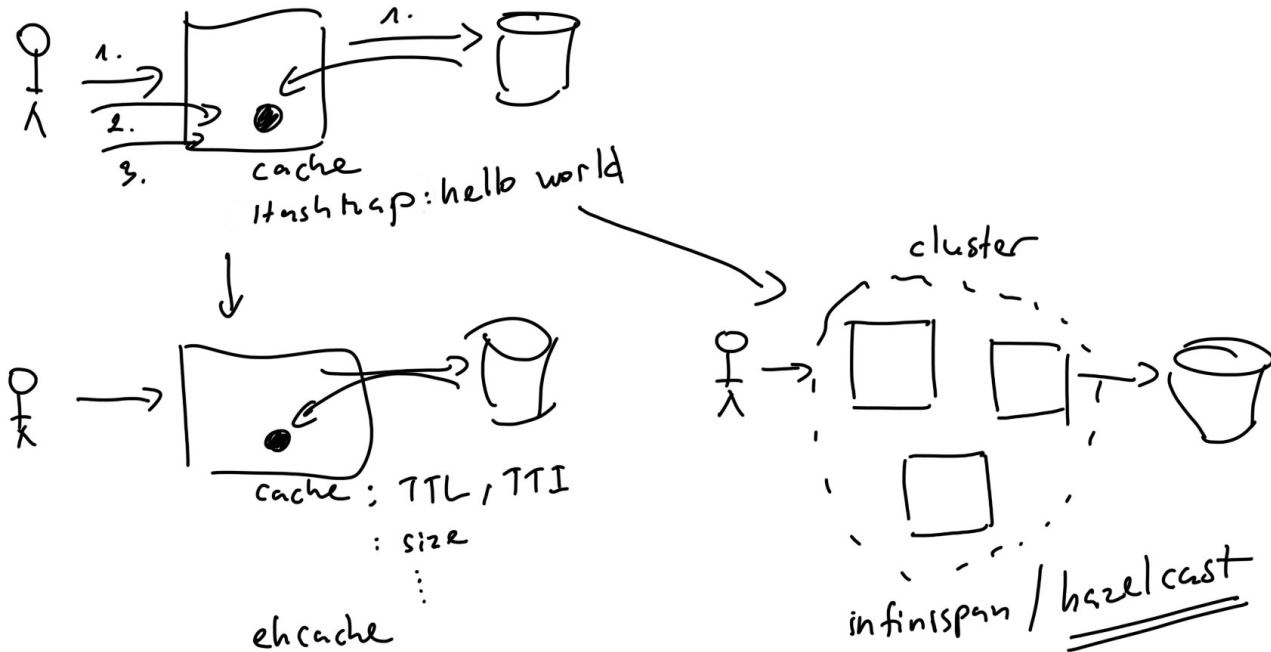
# HQL (JPQL) vs. SQL



Při přístupu na LAZY vazbu uvnitř otevřené transakce  
Hibernate provede SELECT do databáze



# Cache: Map vs. EhCache vs. Hazelcast



# Hazelcast: Embedded vs. Client-Server mód

Hazelcast: embedded:

